Target Tracking Using Sensor-Cloud: Sensor-Target Mapping in Presence of Overlapping Coverage

Subarna Chatterjee and Sudip Misra, *Senior Member, IEEE*

***Abstract*—This work considers tracking of multiple targets using the sensor-cloud infrastructure. As targets enter the coverage zone of multiple sensors, it becomes crucial to schedule sensors and generate distinct clusters of sensors for each target. It becomes challenging to correctly map sensors to targets, in presence of overlapping coverage, to maintain their privacy and correctness of sensed information about the targets. We propose the Dynamic Mapping Algorithm (*S-DMA*) based on the *Theory of Social Choice* for ensuring a 'fair' and unbiased mapping of sensors to targets. The distribution of summation of the preference values of sensors allocated to targets exhibit a standard deviation of 0.71 within 99% confidence interval. This implies that S-DMA maintains uniformity while scheduling sensors for every target.**

***Index Terms*—Sensor-cloud, social choice, target tracking.**

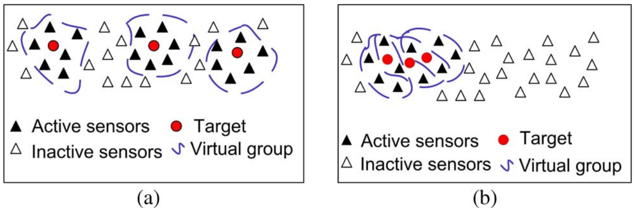
I. Introduction

EXISTING target tracking applications using Wireless Sensor Networks (WSNs) are generally single-user centric, and are configured for a single application [1]. The renderability of customized WSN-based tracking applications is infeasible in traditional WSNs. Organizations that do not own sensor nodes are deprived of access to applications of WSNs deployed in the field. Sensor-cloud is conceived as a potential solution to the sensor management problem of such applications [2].

The cardinal enabling technology behind sensor-cloud is *sensor virtualization*. The physical sensor nodes are dynamically grouped into virtual clusters. An organization requests and obtains sensing service while being unaware of the physical locations of the sensors [3]. The sensed data are transmitted to the end-users over virtual groups, which are then fed to the concerned tracking application. Thus, sensor nodes can be used for *tracking targets as an on-demand service*, rather than as a typical hardware. Such a service can be implemented in real-life such as for surveillance in the military and civilian sectors, traffic control, and health care applications.

A. Motivation

In this paper, we look into the specific issue of multiple target tracking in sensor-cloud, when the mobile targets, belonging to different organizations may come so close that they might fall under the sensing range of one or more physical sensor nodes, thereby leading to overlapping coverage of the targets by the sensor nodes. It is important to generate distinct clusters of sensor-nodes corresponding to each of the targets. The *difficulty* in addressing the problem is that in a conventional WSN, the sensor nodes are typically equipped with some basic analytical and decision-making abilities, and algorithmic processing of data occurs within each sensor node followed by transmission of the target-specific aggregated data. Whereas, in a sensor-cloud environment, sensor nodes are treated as mere sensing units with minimal network management and end user supervision. The raw sensed data are directly transmitted to the sensor-cloud environment. The sensor-cloud service provider aggregates the received data in a target specific manner, before transmitting the aggregated data to the end-users. This introduces the *challenge* to manage distinct sensor clusters for each target. In such a scenario, the problem of sensor-target mapping induces research interest.



(a) (b)

Fig. 1. Local cluster formation (a) Distinct cluster (b) Non-distinct cluster

B. Contribution

The proposed work is not a *trivial extension of the existing works* as prior related works are implemented on conventional WSNs. The *contribution* of this work is to address the above-mentioned issues within sensor-cloud by correctly mapping sensors to their corresponding targets, assuming that a sensor node covers one (Fig. 1(a)) or more (Fig. 1(b)) targets, at a particular time instant. It is required to perform a *primary mapping* of the physical sensor nodes to targets followed by a *secondary mapping* of physical sensor nodes to virtual sensor groups. The work performs a *utility-based* primary mapping within the sensor-cloud environment, simply from the raw sensed data and the previous knowledge about the targets.

We use the *Theory of Social Choice* [4] to *overcome the difficulty* in the problem by applying "fairness" based strategic voting on the society of sensor nodes. We propose and implement *Social-choice based Dynamic Mapping Algorithm (S-DMA)* within a sensor-cloud environment.

II. S-DMA: Social Choice Based Dynamic Mapping Algorithm

We assume that every sensor node is capable of estimating the target coordinates from its sensor reading. We have

Manuscript received October 22, 2013; revised May 3, 2014 and June 22, 2014; accepted June 29, 2014. Date of publication July 8, 2014; date of current version August 8, 2014. The associate editor coordinating the review of this paper and approving it for publication was B. Bellalta.

The authors are with the School of Information Technology, Indian Institute of Technology, Kharagpur 721302, India (e-mail: subarna@sit.iitkgp.ernet.in; smisra@sit.iitkgp.ernet.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LCOMM.2014.2336839

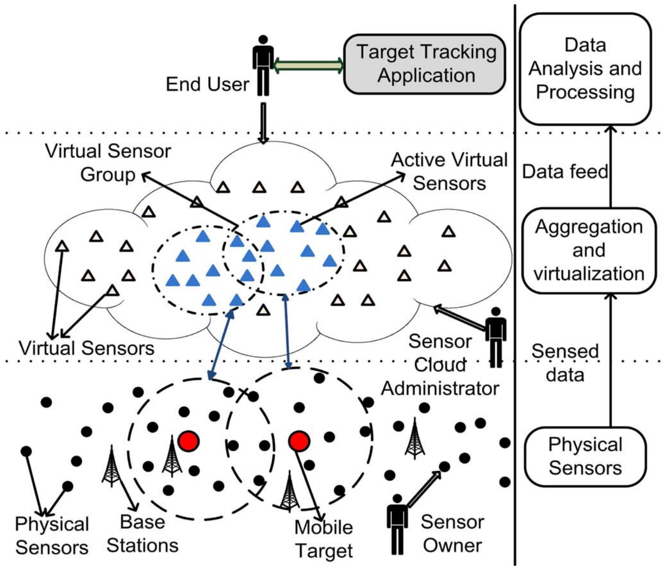


Fig. 2. Sensor-cloud based target tracking.

two symmetric distance matrices $X(1 \ldots N)$ and $Y(1 \ldots N)$, which contain the global coordinates of every node, i, represented as $C(i) = (X[i], Y[i])$. We need to detect the presence of overlapping sensor coverage area involving two or more targets. The proposed architecture is depicted in Fig. 2.

A. Detection of Overlapping Coverage

We assume that n_t is the total number of targets tracked within a sensor-cloud environment. We denote the location coordinates of a detected target t_i at time t by (x_{t_i}, y_{t_i}). The sensor-cloud infrastructure uses the sensed data (x_{t_i}, y_{t_i}) at time $t-1$ and predicts $((x'_{t_i}, y'_{t_i})$, at time t using standard location estimation algorithms. Nodes that are within d distance from the target t_i are activated, where the value of the distance parameter d is predetermined. Thus, we have $\xi((x'_{t_i}, y'_{t_i}), C(j)) \leq d$, for each j, where ξ denotes the Euclidean metric in a 2D plane. The metric $\xi(p, q)$ between two points p and q is expressed as $\xi(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$. Thus, we obtain an array of active sensor nodes from the above relation. To determine the formation of overlapping regions, we need to examine, for every selected node s_i, such that $\xi(C(s_i), t_k) > d$, $k = n_t - 1$. In other words, only for a single target t_j, the inequality must hold true to ensure non-overlapping sensor coverage. However, if an overlap is detected, the scheduling of sensor nodes needs to be governed and steered, accordingly. We assume that a total of n_s overlapped sensor nodes are detected for n_t targets.

Calculation of 'eligibility' factor of a sensor node: Initially, for every possible sensor-target combination, we define a boolean parameter, named 'eligibility' factor. It is the output of a binary function $u(\cdot, \cdot)$, referred to as the 'eligibility' function. The function is expressed as a mapping $u : S \text{ X } T \rightarrow [0, 1]$, where S is the set of overlapping sensor nodes selected for the current set of targets, and T is the current set of targets to be tracked. If ρ_{s_i} is the sensing radius of a node, the mapping u is defined as:

$$u(s_i, t_j) = \begin{cases} 0, & \xi(s_i, t_j) > \rho_{s_i} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

B. Computation of Nodal Preference

Prior to computing the nodal preferences, we introduce a new metric termed *Coverage Contraction Factor (CCF).*

Coverage contraction factor (α_{s_i})*: CCF* is introduced to examine a node's energy content. It computes the residual battery status of a sensor node. If E_{cur} and E_{act} are the current and initial energy levels of a node, we have,

$$\alpha_{s_i} = (E_{act,s_i} - E_{cur,s_i}) / E_{act,s_i} \tag{2}$$

where $0 \leq \alpha_{s_i} \leq 1$. As we base our approach on the *Theory of Social Choice* [4], every active sensor node has its own preference of targets, articulated by means of a linear ordering. Preference P_i and indifference I_i of a node i are the symmetric and asymmetric components of the relation, respectively [4]. Thus, $t_a P_i t_b \nRightarrow t_b P_i t_a$. Also, $t_a I_i t_b \Rightarrow t_b I_i t_a \Rightarrow t_a \equiv t_b$.

We now focus on evaluating a node's ordering of preferences for each target of interest at time t. After the data from the physical sensor nodes are transmitted, nodal preferences are evaluated on servers of the sensor-cloud infrastructure. We design a utility function Ψ for every sensor-target pair. We have,

$$\Psi(s_i, t_j) = \begin{cases} \frac{\lambda_1}{\alpha_{s_i}} + \lambda_2 \frac{\rho_{s_i}}{\xi(s_i, t_j)} : & \alpha_{s_i} \neq 0 \\ \mathbb{B} + \lambda_2 \frac{\rho_{s_i}}{\xi(s_i, t_j)} : & \alpha_{s_i} = 0 \end{cases} \tag{3}$$

where $\mathbb{B}$ is a large integral value, and λ_1, λ_2 (when $\lambda_1 < \lambda_2$) are the weighted system-modeled coefficients. Nodal ordering of preferences of targets are based on a *preference* value Θ, which is defined as:

$$\Theta(s_i, t_j) = \Psi(s_i, t_j) \times u(s_i, t_j) \tag{4}$$

If a node is not *eligible* for tracking a particular target, the *preference* value is zero. Having calculated the *preference* value for every sensor-target pair, each sensor node then creates its own ordering of choices. For a sensor node s_i, $\Theta(s_i, t_a) > \Theta(s_i, t_b) \Rightarrow t_a P_{s_i} t_b$, $\Theta(s_i, t_a) = \Theta(s_i, t_b) \Rightarrow t_a I_{s_i} t_b$. However, the preference ordering for every sensor node should be *complete* and *transitive* [4]. Hence, it implies,

$$(t_j P_{s_i} t_k) \vee (t_j I_{s_i} t_k), \quad \forall\, j, k \in T \tag{5}$$

$$(t_j X_{s_i} t_k) \wedge (t_k X_{s_i} t_l) \Rightarrow t_j X_{s_i} t_l, \quad \forall\, j, k, l \in T \tag{6}$$

where $X_{s_i} = \{P_{s_i}, I_{s_i}\}$. Equations (5) and (6) ensure the *completeness* and *transitivity* axioms, respectively. After obtaining the preferences of every node at time t, we obtain a matrix $\Theta_{net}[1 \ldots n_s][1 \ldots n_t]$ for the entire network. We now define some relevant terms.

Definition 1: A preference ordering R_{t_i} for a particular target t_i is the set of preference values of the different sensor nodes casted for t_i, i.e., $R_{t_i} = \{\Theta_{s_1,t_i}, \Theta_{s_2,t_i}, \ldots, \Theta_{s_{n_s},t_i}\}$.

Definition 2: A preference profile $\mathcal{P}$ is the set of potential preferences, i.e., $\mathcal{P} = \{R_{t_1}, R_{t_2}, \ldots, R_{t_{n_t}}\}$.

C. Social Choice Aggregation

Once the preference profile is established, we embark on the *Social Aggregation Function (SAF)* and *Social Choice Function (SCF)*. The SAF is defined as a mapping $F : \mathcal{P}^{n_s!} \rightarrow \mathbb{R}^{n_t \times n_{th}}$, i.e., F maps a preference domain to a mapping matrix M, $M[i][j]$ denotes the allocation of the jth sensor node to target t_i, n_{th} is the threshold value for the maximum number of sensor nodes that can be allocated to a target. SCF f is defined as a mapping $f : \mathcal{P} \times T \rightarrow S$, i.e., given a preference profile and a particular target, a particular sensor node or the social choice winner s_{win} can be mapped to the target based on the choice of the society.

$$F(\mathcal{P}) = F\left(R_{t_1}, R_{t_2}, \ldots, R_{t_{n_t}}\right) = M, \quad f\left(R_{t_i}\right) = s_{win} \tag{7}$$

In this work, we have assumed $n_t \ll n_s$. Multiple iterations are performed on random ordering of targets till all sensor nodes are allocated. For the selection of a 'fair' winner, as per *Arrow's Impossibility Theorem* [5], we merge the positive effects of *Plurality Voting* [4] and *Borda's algorithm* [4] with the proposed algorithm, as presented in Algorithm 1. For t_i, we initially formulate the society mean μ_{soc}, expressed as,

$$\mu_{soc} = \frac{\sum_{\forall s_j \in S} \beta_{s_j} \times \Theta(s_j, t_i)}{\sum_{\forall s_j \in S} \beta_{s_j}} \tag{8}$$

where $\{\beta_{s_j} \times \Theta(s_j, t_i)\}$ denotes the is the social preference order. $r(s_j, R_{t_i})$ is the positional value[1] of a voter in the profile of a target, expressed as, $\beta_{s_j} = n_s - r(s_j, R_{t_i})$. The winner node s_{win} is obtained by,

$$s_{win} = M[i][win] = \min_{\forall s_j \in S} \left|(\mu_{soc} - \Theta_{s_j, t_i})\right| \tag{9}$$

The winner node s_{win} can be considered as the *Plurality* winner, as its score for t_i is the closest to the society mean, thereby earning the highest ability to win the target.

Algorithm 1 S-DMA

```
Inputs:
  • Set of mobile targets: $T$.
  • Set of sensor nodes with overlapping coverage: $S$.
Output: A mapping matrix $M[1 \ldots n_t][1 \ldots n_{th}]$.
1: $\forall s_i \in S, \forall t_j \in T$, Compute $\Theta_{net}$
2: while $(\exists s_i \in S) \wedge (s_i \neq M[p][q]), \forall p \in T, \forall q \in S$ do
3:   Generate a random ordering of targets $\hat{T}$
4:   for $j = 1$ to $n_t$ do
5:     $M[j][win] = s_{win} = f(R_{\hat{T}_j})$
6:     Remove $s_{win}$ from $S, \mathcal{P}$
7:   end for
8: end while
```

III. Analytical Results

Proposition 1: The worst case asymptotic computational complexity of S-DMA and communicational complexity for a single target are $O(n_s \times n_t)$ and $O(n_s^2)$, respectively.

Proof: For n_t number of mobile targets, step 1 of *S-DMA* is computed in $n_s \times n_t$ time. Steps 3, 5, and 6 of Algorithm 1 is executed in constant time c_1. Steps 4 through 7 takes $O(n_t)$. Thus, we have,

$$T(n_t) = n_s n_t + T(n_s - 1) + c_1 n_t, \quad T(1) = c_2$$

Simplifying, we get, $T(n_t) = O(n_s \times n_t)$. For communication load, every node, s_i, communicates with the sensor-cloud through multi-hop route. The number of hops is $O(i-1)$. Communication load C for a single target involving n_s overlapping sensors can be expressed as $C(1) = \Theta(1)$, $C(n_s) = \sum_{i=2}^{n_s} O(i-1) \simeq O(n_s^2)$. This completes the proof. ■

Lemma 1: S-DMA satisfies non-dictatorship.

Proof: An SCF is dictatorial if $\exists s_i : t_a P_{s_i} t_b \Rightarrow t_a P_{s_j} t_b$, $\forall s_j \in S$ [5]. But in *S-DMA*, $\forall s_i \in S$, we have, $\Theta_{s_i,\cdot}$. Also, $\nexists s_i : t_a P_{s_i} t_b \Rightarrow t_a P_{s_j} t_b$, $\forall s_j \in S$. For any target t_i, if $f(R_{t_i}) = s_{win}$, $\nexists s_i$, such that, $f(R_{t_i}) = s_j$, where $S = S - \{s_i\}$, or $S = S + \{s_i\}$, $s_i \neq s_j$. Thus, *S-DMA* is non-dictatorial. ■

Lemma 2: S-DMA satisfies the Independence of Irrelevant Alternatives.

Proof: Independence of Irrelevant Alternatives (IIA) claims that the internal ranking between two alternatives is independent of a third alternative [5]. In *S-DMA*, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two preference sub-profiles containing $\hat{S}$ sensor nodes, and let t_i an t_j be two target alternatives, such that $\hat{S} \subset S$, $t_i \succeq_{\mathcal{P}_1} t_j$ and, $t_i \succeq_{\mathcal{P}_2} t_j$. Then, *S-DMA* concludes that $\forall s_i \in \hat{S}$. Thus, $\Theta_{s_i,t_i} \geq \Theta_{s_i,t_j} \Rightarrow \sum_{\forall s_i \in \hat{S}} \Theta_{s_i,t_i} \geq \sum_{\forall s_i \in \hat{S}} \Theta_{s_i,t_j} \Rightarrow t_i \succeq_{\hat{S}} t_j$. This concludes the proof. ■

Corollary 1: S-DMA dissatisfies the Pareto Axiom (P).

Explanation: From Arrow's Impossibility Theorem [5], we obtain that no aggregation function can simultaneously satisfy non-dictatorship, IIA and P. Thus, from Lemma III.1 and III.2 we infer that *S-DMA* dissatisfies P.

Theorem 1: S-DMA tends to select the Condorcet winner.

Proof: We assume the sensor node preferences as (t_a, t_b, t_c), (t_b, t_a, t_c), (t_b, t_c, t_a), (t_c, t_b, t_a), (t_c, t_a, t_b), and (t_a, t_c, t_b). Let t_a be the Condorcet winner (through pairwise voting) for some sensor node s_i. We must have, $\Theta_{s_2} + \Theta_{s_3} + \Theta_{s_4} + \Theta_{s_5} \leq \Theta_{s_1} + \Theta_{s_6}$. Assuming the correctness of a Condorcet winner, we consider a hyper-plane divided into distinct sensor node regions. The normal to the correct side is obtained as $N_1 = (1, -1, -1, -1, -1, 1)$. In *S-DMA* let the positional significance be (2,γ,0). From (9), we get, $\gamma\Theta_{s_2} + \gamma\Theta_{s_5} \leq 2\Theta_{s_1} + 2\Theta_{s_6}$. We obtain $N_2 = (2k, -\gamma k, 0, 0, -\gamma k, 2k)$, assuming $\Theta_{s_i,t_a} = k, \quad \forall s_i \in S$. If ϕ is the angle between N_1 and N_2, $\cos(\phi) = (N_1.N_2/|N_1||N_2|) = (2\gamma + 4)/(\sqrt{6}\sqrt{2\gamma^2 + 16}) = h(\gamma)$. Thus,

$$\frac{dh(\gamma)}{d\gamma} = \frac{1}{\sqrt{6}} \frac{2\sqrt{2\gamma^2 + 16} - 4(2\gamma^2 + 16)^{-\frac{1}{2}}(2\gamma + 4)}{(2\gamma^2 + 16)} \tag{10}$$

The angle between the normals should be minimized to obey Condorcet criterion. Evaluating, $dh(\gamma)/d\gamma = 0$, we get $\gamma \rightarrow 1$. Thus, correct positional value is assigned by following *S-DMA*. This concludes the proof. ■

Proposition 2: S-DMA respects Plurality voting.

Proof: Plurality voting selects a winner agent, which obtains the highest score of the society. From (9), we find that the winner node s_{win} satisfies $|(\mu_{soc} - \Theta_{s_{win},t_i})| \rightarrow 0 \Rightarrow \Theta_{s_{win},t_i} \rightarrow \mu_{soc}$. Thus, $\Theta_{s_{win}}$ respects μ_{soc}, which is a society parameter. Hence, s_{win} is also the *Plurality* winner. ■

We now present some of the experimental results. To validate the correctness of *S-DMA*, we assume a *uniform random deployment* of 250 sensor nodes ($\rho_{s_i} = 200$ m, $C_{s_i} \geq 2\rho_{s_i}$) in an area of 1 km × 1 km, C_{s_i} being the communication range. Fig. 3 shows that two targets enter the zone, and move close to each other. *S-DMA* clearly outperforms the existing

[1] Positional significance of a node can be viewed as its Borda score. However, it is not explicitly termed as 'Borda' score, as the Borda score is ideally applicable to candidates instead of voters.

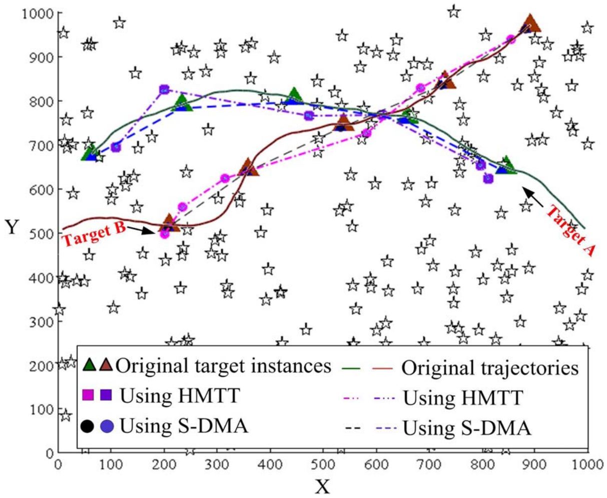


Fig. 3. Projection of *S-DMA* against HMTT.

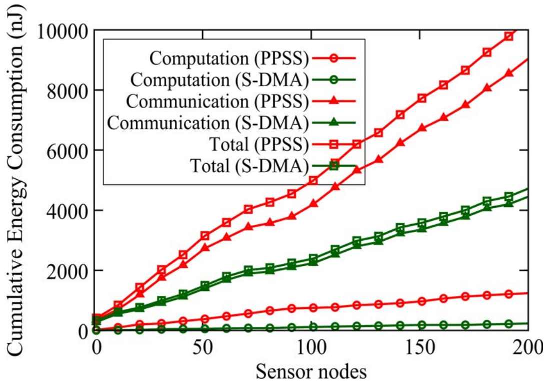


Fig. 4. Comparison of energy consumption.

algorithm—Hierarchical Markov Decision Process (HMDP) for target tracking (HMTT) [6], in terms of tracking accuracy. Unlike [6], *S-DMA* proposes a "fair" sensor-target mapping, which improves the tracking accuracy especially in situations of overlapping coverage of sensors. Assuming the communication and processing energy as 40 nJ/bit and 10 nJ/bit, respectively, and subjecting the algorithms to identical sensing phenomenon, Fig. 4 clearly shows that, unlike Probability-based Prediction and Sleep Scheduling protocol (PPSS) [7], S-DMA exhibits a low energy consumption for computation as the processing and evaluation is mainly executed at the sensor-cloud end. Further, in PPSS, multi-hop communication within the network and data transmission to a data center contributes for the overall communication energy. On the contrary, inter node communication is negligible in S-DMA, multi-hop transmission being the main component for energy consumption. This conserves the total energy appreciably. To enhance the understandability, Fig. 5 depicts the combined impact of positional significance and the preference values on the collective preferences of a target for 10 sensor nodes. The preference profile curve is aligned to the secondary y axis. Fig. 6 demonstrates the difference of magnitude of the normal mean from the social mean. The experiment is executed to compute the standard deviation of the summation of preference values for the sensors assigned to each target. The mean of the standard deviations over 100 iterations was found close to 0.71 with a 99% confidence interval. This suggests that the proposed algorithm is unbiased to targets and maintains uniformity while mapping.

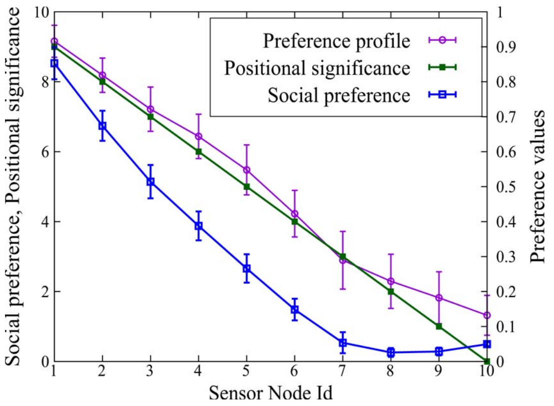


Fig. 5. Target specific nodal preferences.

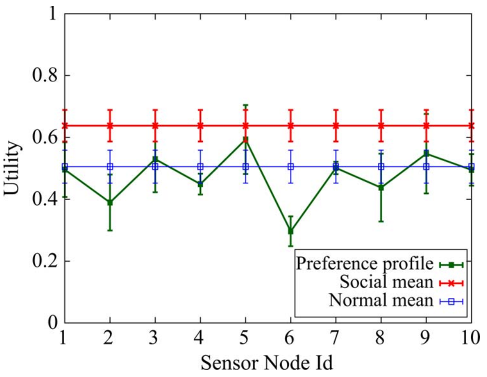


Fig. 6. Analysis of R_{t_i}.

IV. Conclusion

The proposed algorithm, *S-DMA*, ensures the best possible allocation of sensors to targets. However, there can be challenges if two adjacent sensor nodes are heterogeneous with respect to their sensing types as multi-hop communication in such scenario will require protocol standardization. Our future work will focus on extending the current problem in context of such heterogeneous sensor nodes.

References

[1] E. Onur, C. Ersoy, and H. Delic, "Analysis of target detection probability in randomly deployed sensor networks," *IEEE Commun. Lett.*, vol. 11, no. 10, pp. 778–780, Oct. 2007.

[2] A. Alamri *et al.*, "A survey on sensor-cloud: Architecture, applications, approaches," *Int. J. Distrib. Sensor Netw.*, vol. 2013, pp. 917923-1–917923-18, Nov. 2012.

[3] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure—Physical sensor management with virtualized sensors on cloud computing," in *Proc. 13th Int. Conf. NBiS*, Sep. 2010, pp. 1–8.

[4] D. Bouyssou, T. Marchant, and P. Perny, *Decision-Making Process: Concepts and Methods*. Hoboken, NY, USA: Wiley, Jan. 2010.

[5] J. Geanakoplos, "Three brief proofs of arrow's impossibility theorem," in *Cowles Foundation for Research in Economics*. New Haven, CT, USA: Yale Univ. Press, 2001.

[6] W. L. Yeow, C. K. Tham, and W. C. L. Wong, "Energy efficient multiple target tracking in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 918–928, Mar. 2007.

[7] B. Jiang, B. Ravindran, and H. Cho, "Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 4, pp. 735–747, Apr. 2013.